

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Control Optimization Using MATLAB

Patic Paul Ciprian, Duta Luminita and Pascale Lucia
Automatics, Informatics and Electrical Engineering Department
Valahia University of Targoviste
Romania

1. Introduction

The automatic modelling system using MATLAB-Simulink software package applies the Cohen-Coon method for determining the closed loop PID parameters and plotting the system response curve. Implementing the same system in a closed loop PID adjustment using LabVIEW software package, requires, at first, an additional procedure, namely, the use of transfer functions in Z domain, which means the use of a sampling signal. The study in this paper intends to determine the parameters of a PID regulator using two software packages - MATLAB-Simulink and LabVIEW. As a mathematical pattern, a DC machine has been used. To exemplify regulators such as PID, LQR (Linear Quadratic Regulator) and PI-MIMO (PI - Multiple Input Multiple Output) were used. The application is easily implemented, the inverted pendulum being used to emphasize the results. The closed loop system and its evolution over time are easily implemented. This software package offers an easy user interface, which means it is convenient to introduce the programme functions (Nițu, C., et al., 1974).

MATLAB is a tool with high performance in technical estimations. It integrates the calculation, visualization and programming in an easy to use environment where problems and solutions are expressed in familiar mathematical notation. Using the MATLAB software package has the following main purposes: mathematical calculation, the development of algorithms, data acquisition, modelling, simulation and prototype development, visualization, exploration and data analysis, scientific graphics and engineering, and application development of a graphical user interface. MATLAB is an interactive system whereby the basic element to define data is the string that does not require sizing (Andrei, H., et al., 1999; Călin, S., 1970; Călin, S., et al., 1979, 2002; Dumitrache, I., 1971; Papadache, I., 1975).

With LabVIEW virtual instruments are built (VI), having the appearance of instruments or physical systems. Virtual instruments have an interactive user interface front panel - and with one part for the programmer - block diagram. For identification and use in other applications, each virtual instrument has an icon with specific entries and exits (Bishop, R., 2010).

Automatic systems modelling using the MATLAB-Simulink software package applied the Cohen-Coon method for determining the closed loop PID parameters and plotting the system response curve. The limit of the over adjustment was found to be 20%, the timing of the transitional regime - 3 seconds, and the maximum response time was 10 seconds. Also, one represented the chart of the evolution over time, in a closed loop system, to unit step input.

This software package offers an easy user interface, which means that it is convenient to introduce the programme functions. The application has implemented the conversion done from a field of definition of transfer functions to another (ex. obviously $c2d$ function transforming the transfer function of s field in Z domain, using a sampling period by the user), and includes the Simulink tool that greatly facilitates the user's work as the user wants to consider in detail the operation of each block and many other features in the field of automation and theory systems.

MATLAB integrates the calculation, visualization and programming in an easy to use environment where problems and solutions are expressed in familiar mathematical notation.

Automatic systems modelling using the MATLAB-Simulink software package applied the Cohen-Coon method for determining the closed loop PID parameters and plotting the system response curve. This software package offers an easy user interface, which means that it is convenient to introduce the programme functions.

This paper intends to determine the parameters of a PID regulator using two software packages - MATLAB-Simulink and LabVIEW (Mahesh, L., et al., 1998).

To exemplify, PID, LQR (Linear Quadratic Regulator) and PI-MIMO (PI - Multiple Input Multiple Output) regulators types were used.

In the case of the LQR regulator, the Control System Toolbox was necessary since it is an extended version of the problem from the Simulink demo file.

Using LQG / LTR techniques, one can design a Kalman state estimator and a corresponding amplifier regulator for the linear system, by adding an integrator to ensure a zero steady error (Gibson, J., 1967; Ionescu, V., 1975).

An initial design regulator using the LQG / LTR methods is designed starting from a linear application.

The third problem of control design involves designing a centralized PI-MIMO regulator for a turbine engine of LVI00 fuel. The application is modelled as a system with two inputs, two outputs and a minimum of five phase states. Inputs are fuel flow and variable area turbine nozzle.

The work from this paper intended to determine the parameters of a PID controller using two software packages, namely MATLAB-Simulink and LabVIEW.

A comparative analysis of the features of the two programmes - MATLAB-Simulink and LabVIEW to determine the same automatic performances of the system - indicates that the MATLAB-Simulink software is more efficient. It has different features from its libraries already implemented, with coverage areas of automation and systems theory. LabVIEW is a software package dedicated to particular virtual instrumentation and graphical applications in time (Bishop, R., 2010).

Passing to study cases, first of all, one tries to present the classical case of the inverted pendulum and, also, the modelling of the robotic arm. We compared the results obtained using MATLAB-Simulink and LabVIEW software. The equations of the model are presented and the modelling results are given comparatively.

2. PID regulator

At first, the problem of control design is shaped by the transfer application as having a single input and a single output (Single Input Single Output, SISO) of third order, where $a_1 = 43$ and $a_1=3$ with an approximation of ± 0.7 and a non-linear saturation ± 1.5 .

In addition, due to design tolerances, the current dynamics presents suggestive variations from the base model. Specifically, the denominator coefficient a_2 varies between 40 and 50 and coefficient a_1 between half and 1.5 times the nominal value of 3 (Franklin, G., et al., 1987; Potvin, A. F., 1991).

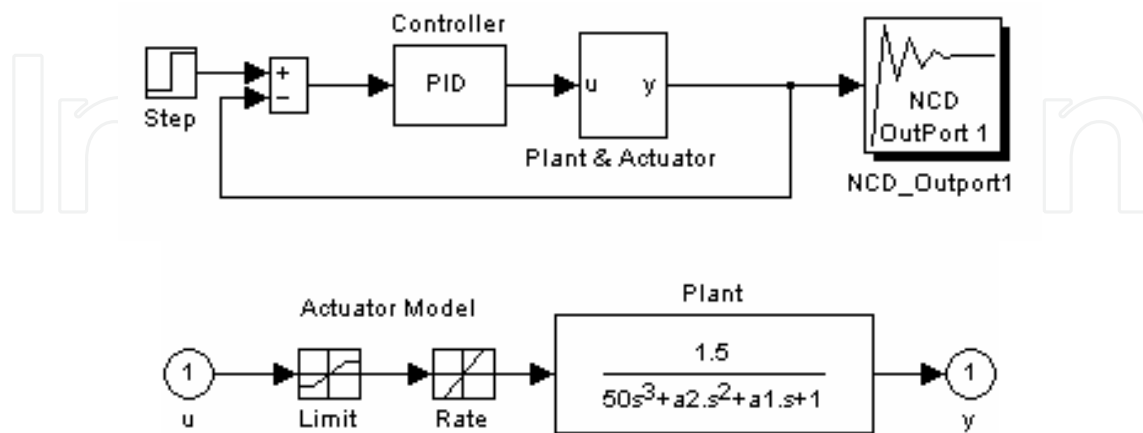


Fig. 1. NCD block attached to the result of the plant

$$G(s) = \frac{1.5}{50s^2 + a_2s^2 + a_1s + 1} \quad (1)$$

A PID regulator is designed so that closed-cycle system may meet the following specifications for tracking:

- Maximum oscillation 20%;
- Up to 10 seconds for the propagation;
- Maximum response time of 30 seconds (Franklin, G., et al., 1987; Potvin, A. F., 1991).

This results in the answer that the closed-cycle is resistant to uncertainty in the installation's dynamics.

The Simulink `nccdemo1` system contains the application and the control structure. It is noted that non-linearity and saturation speed are included in the model plant. An input step drives the system. An NCD block is attached to the outlet installation (Figure 1) because it is the signal which will be restricted. On checking the System's Parameters dialog box one can notice that each simulation lasts 100 seconds.

Tuneable and uncertain variables are initialized. The uncertain variables a_1 and a_2 are initialized at nominal values of 40 and 3 respectively. Tuneable parameters K_p , K_i and K_d are initialized at 0.63, 0.05, 2.02 respectively. These values result from the use of the Ziegler-Nichols method for PID regulators.

The Ziegler-Nichols method for PID regulators can be summarized as follows:

- The entire amplification and derivative is established to zero. Then, all these are increased proportionally, until the system becomes unstable;
- This amplification is defined as K_u , the measure of the oscillation period is P_u ;
- The coefficients are established one by one:

$$K_p = \frac{3 * K_u}{5}, \quad K_i = \frac{6 * K_u}{5 * P_u}, \quad si \quad K_d = \frac{3 * K_u * P_u}{50} \quad (2)$$

Then, the limitations of time are defined. Upper and lower restriction limits define oscillation, propagation time and response time (Franklin, G., et al., 1987; Potvin, A. F., 1991).

Note that the uncertainty in the parameters a_1 and a_2 is defined and that the restrictions are applied to the installation only nominally. After running optimization, the time, the cost function evolution and the final values for tunable parameters vary depending on computer's performance. Optimization should provide a regulator to satisfy any restrictions (National Instruments; MathWorks; MathWorks User's Guide, 2003).

The lower and superior limits are restricted and the optimizing process is started with uncertainty. One can consider that these restrictions cannot be satisfied, but the result shows a maximum violation of the restriction under the value of 0.01.

One can experiment by moving the limits of restriction to achieve better performances of the system. For example, one can reduce the time spread or decrease the restrictions of oscillation (Figure 2.)

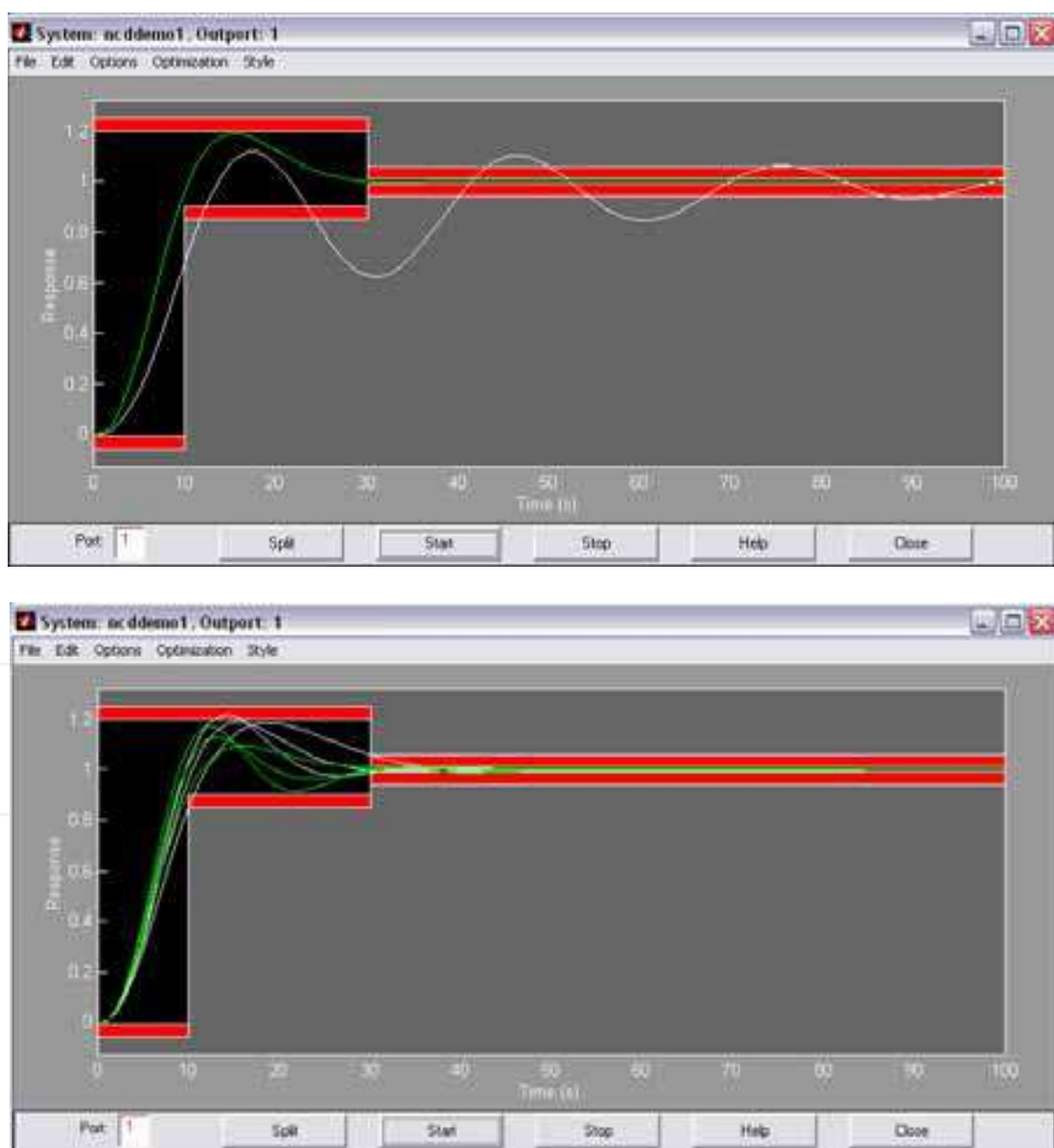


Fig. 2. Moving the limits of restriction to achieve better performance of the system

3. LQR regulator with feed-forward controller

The second problem requires a Control System Toolbox since it is an extended version of the problem from the Simulink demo *lqgdemos* file.

The SISO application can be modelled as a linear ranking 4 system with an enlarging saturation by ± 5 and a non-linear limit of error of ± 10 .

The equations are (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003):

$$\dot{x} = \begin{bmatrix} -1.0285 & 0.9853 & -0.9413 & 0.0927 \\ -1.2983 & -1.0957 & 2.8689 & 4.7950 \\ 0.1871 & -3.8184 & -2.0788 & -0.9781 \\ 0.4069 & -4.1636 & 2.5407 & -1.4236 \end{bmatrix} x + \begin{bmatrix} 0 \\ 6.6389 \\ 0 \\ 0 \end{bmatrix} u = Ax + Bu \quad (3)$$

$$y = [-1.7786 \quad 1.1390 \quad 0 \quad -1.0294] = Cx$$

They define the nominal plant model. One allows the installation's matrix to vary between half to two times of its nominal value.

Using LQG / LTR techniques, one can design a Kalman state estimator and a k amplifier regulator for the linear system. Add an integrator to ensure a zero steady error.

To achieve an increased time response one adds a feed-forward amplifier (FF).

In the demo Simulink *Iqgopt* system, the control parameters k and FF are granted by 'the method presented above (National Instruments; MathWorks; MathWorks User's Guide, 2003).

These parameters can be accorded using the NCD Blockset.

In particular the control parameters k and FF are granted, so that the closed-cycle system may meet the following specifications:

- Maximum oscillation - 20%;
- Propagation time - one second;
- Response time - three seconds (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

The Simulink system contains the application and the control structure below:

After starting the system, it is noted that the non-linearity error (± 10) and the saturation (± 5) are included in the model's installation.

Using the From Workspace block, one introduces a step that goes from zero to one in one second. An NCD block is attached to the result of the installation because there is a restricted signal. Checking the System's Parameters one observes that each simulation always lasts 10 seconds.

Tunable and uncertain variables are initialized. Domain restrictions for this demonstration are defined. Upper and lower restriction limits on the oscillation, propagation time and response time are defined. As described above, an initial design regulator using the LQG / LTR methods is designed starting from a linear application. For non-linear control optimization the feed-forward amplifier FF and regulatory matrix amplifier k are tunable (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

Note how the installation (reaction) matrix A is defined and also that the optimization restrictions are applied only on nominal installation (reaction).

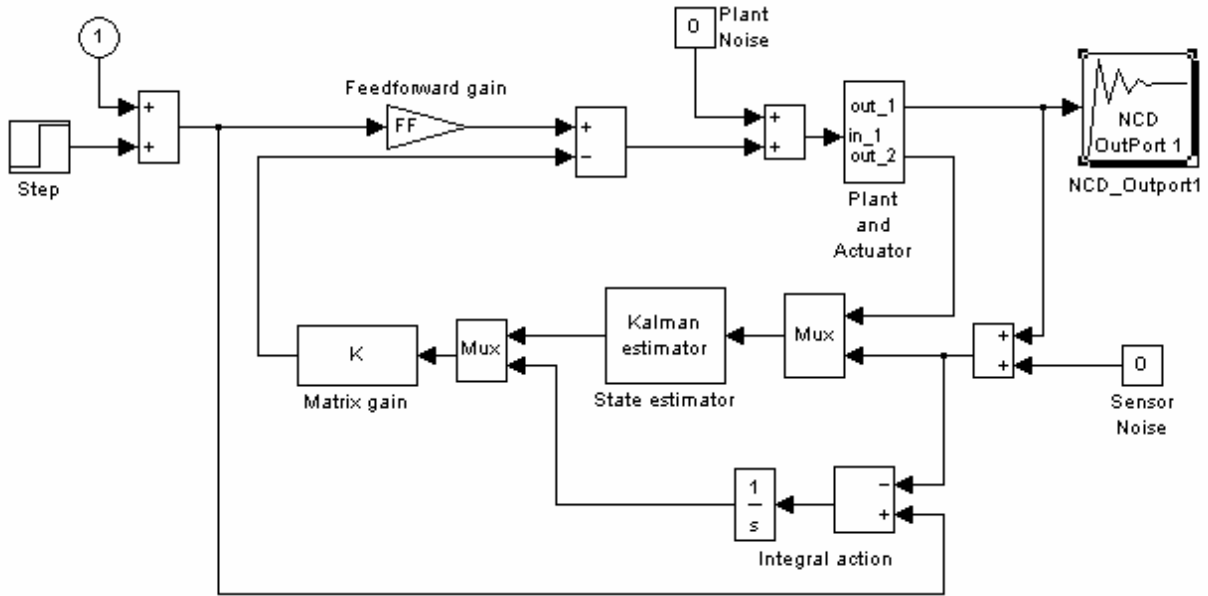


Fig. 3. NCD block attached to the result of the plant

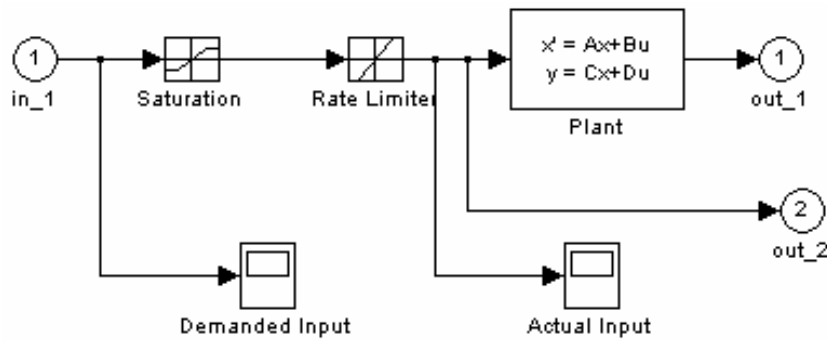


Fig. 4. Plant & Actuator

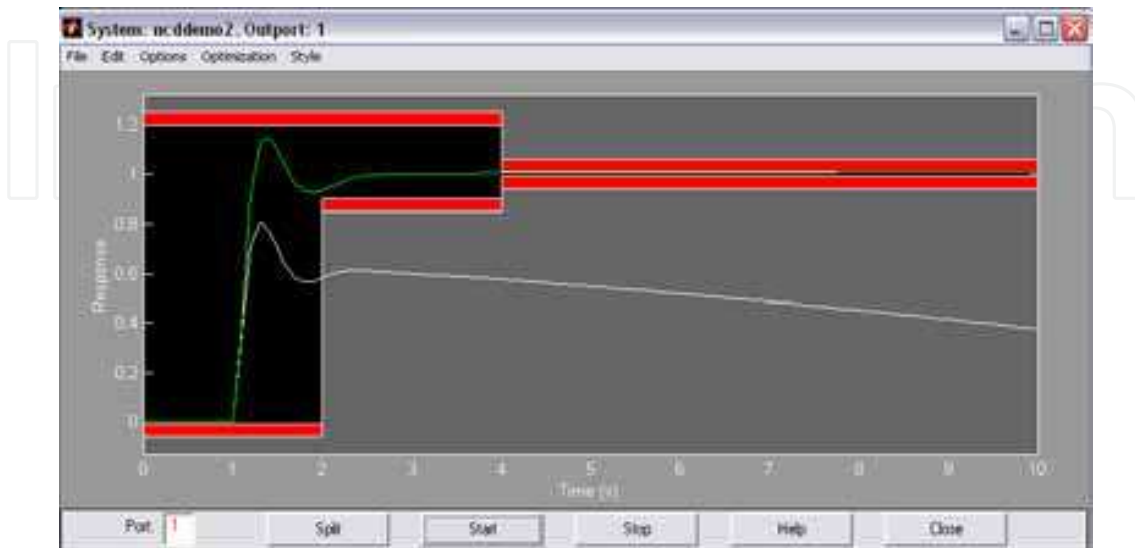


Fig. 5. Moving the limits of restriction to achieve better performance of the system

After the optimization starts, one monitors the response evolution in time. The optimization time, the cost function evolution and final values for tunable parameters vary depending on computer (Savant, C. J., 1967; Sprânceană, N., et al., 1978).

Optimization should produce a regulator which satisfies any restrictions.

One must always restrict the upper and lower limits and start the optimization with uncertainty. Considering that these restrictions can be satisfied, the result shows a maximum violation of the restriction under 0.01 (1%).

One can experiment by moving the limits of restriction to achieve better performances of the system; for example, reducing the propagation time or using restrictions on oscillation (Figure 5).

3.1 Results and simulation after using LQR

Simulation using the optimal LQR has been implemented. Figure 6 shows the unit step system response. As shown in the figure, the displacement reaches its final value in 3×10^{-6} sec., gives zero tracking error and the system has better stability. In comparison to the results obtained previously, it is quite clear that the use of LQR triggers a much better response (Jalili-Kharaajoo, M. and Dahastani, A., 2003; Roshan N., 2004).

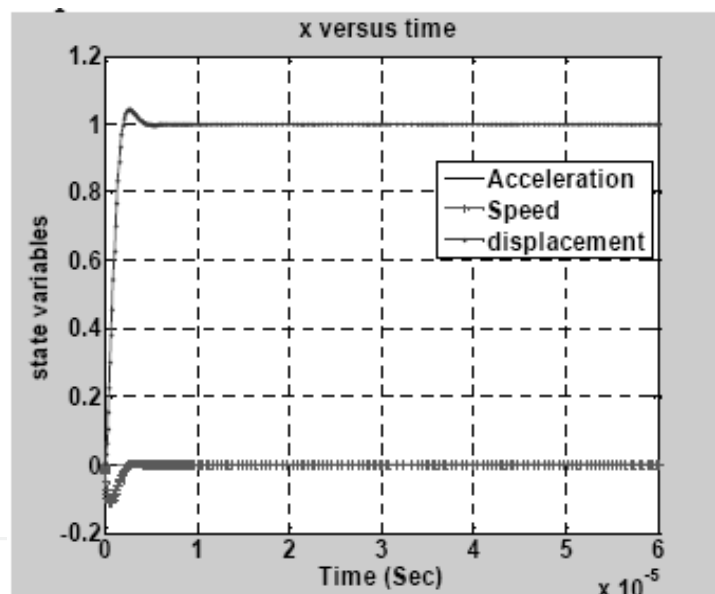


Fig. 6. State variables versus time for unit step input tracking

Figure 7 and Figure 8 show unit ramps and the unit acceleration system response. As shown from the figure the displacement reaches its final value in a few microseconds and gives zero tracking error, which means much better sensitivity and accuracy for the micro accelerometer compared to previous approaches.

The speed of reaching the final values depends on choosing the values of matrix Q , as choosing high values of Q means having faster response for any input signal and having better stability.

Finally, for all inputs, the mass speed when using the LQR is faster and almost has no oscillations, for example, for unit step input the average speed is 1.82×10^{-3} m/sec., compared to 1.71×10^{-4} m/sec. as described by some authors. That is, the mass reaches its final position faster with no oscillations.

Using LQR, has increased and improved the accelerometer bandwidth, stability, accuracy and response time (Ibrahim, H. E. A., 2010; Jalili-Kharaajoo, M and Dahastani, A., 2003).

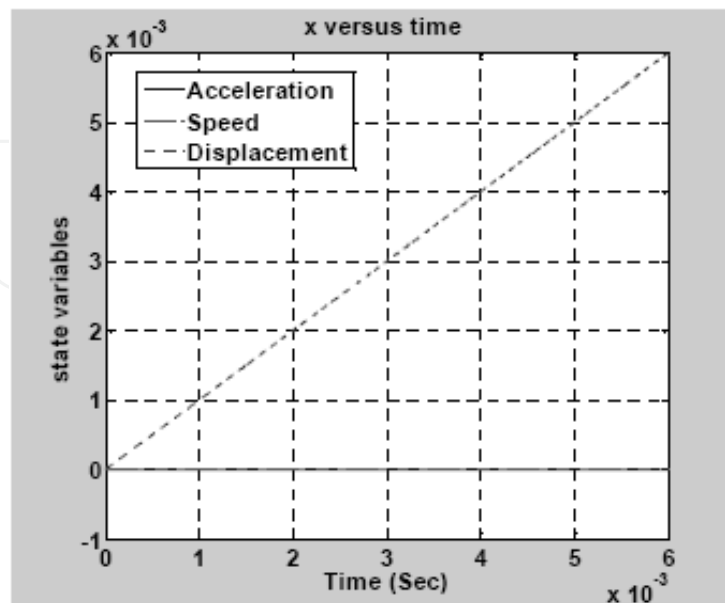


Fig. 7. State variables versus time for unit ramp input tracking

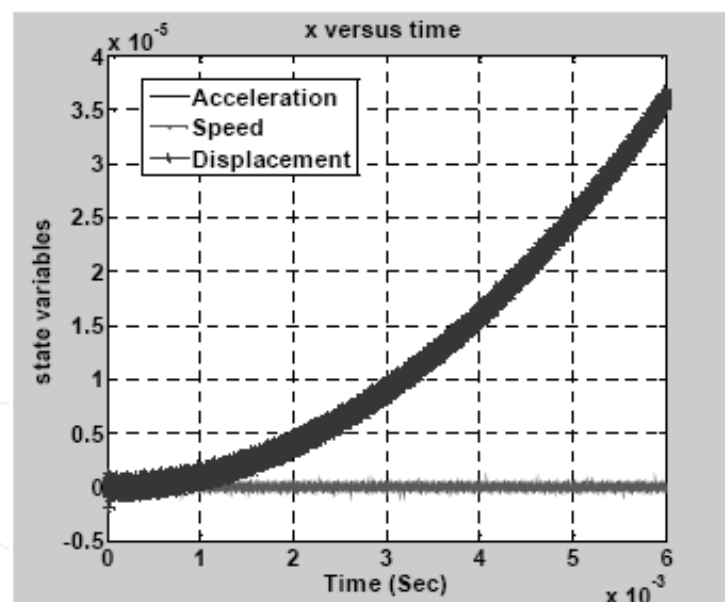


Fig. 8. State variables versus time for unit acceleration input tracking

4. PI-MIMO regulator

The third problem of control design involves designing a centralized PI-MIMO regulator for a turbine engine of LVI00 fuel. One models the application as a system with two inputs, two outputs and a minimum of five phase states. The inputs are fuel flow and variable area turbine nozzle. The outputs are speed coil generator fuel and temperature. The five stages are speed spool generator, the output power, temperature, the driving level of the fuel flow

and the driving level of the turbine nozzle (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

Saturation non-linearities exist in the system as limited efforts and maximum temperatures. These non-linearities can be included in the problem formulation as in previous examples. Also, the demonstration plant will exaggerate uncertainty of installation.

More precisely, one permits a variation of a matrix A between half and twice its nominal value.

We want to design a centralized controller 2-by-2 PI for the application so that the system closed-cycle track meets the following specifications:

- Maximum propagation delay time of one second;
- 0 oscillation in the first channel and less than 10% in one second;
- Less than 5% across the channel coupling.

The Simulink system contains the software and control structure below (Figure 9).

To open the system this can be written in MATLAB prompt or double-click on the NCD block 3 demo Simulink system.

One model the PI regulator as a state-space system with the value 0 for matrix A and the identical matrix B.

The C and D matrix are variable tuneable K, and Kp, for a total of eight tuneable variables.

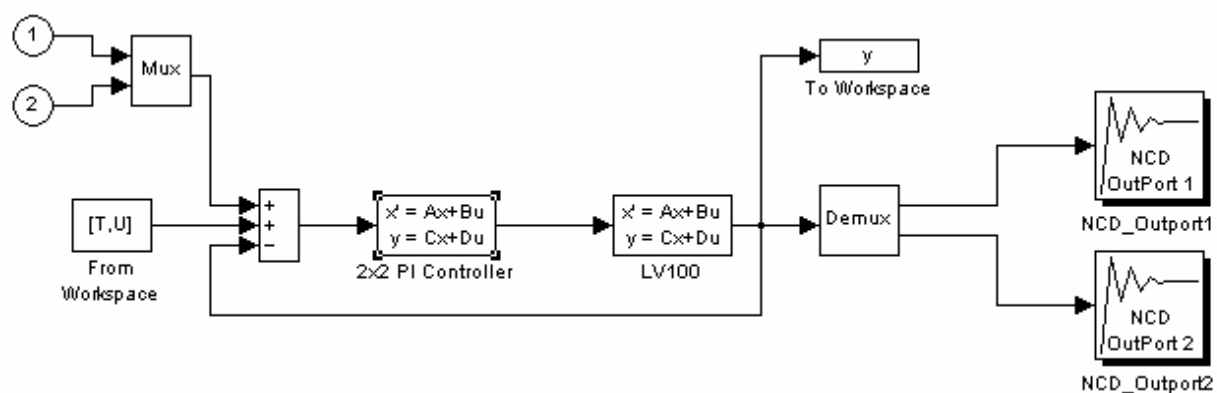


Fig. 9. Control structure of the system

One can note that there are two optimization of NCD blocks that can be displayed simultaneously (Mejhed, N. H., et al., 2005).

The approach suggested for MIMO regulator design requires the sequencing of input steps. When the first channel comes in, the first result should follow up the step and the other channels must reject the signal. When the second channel comes in, the second signal must follow up the step and the other channels must reject the signal etc. (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

Note that we used the From Workspace block to enter the sequential steps inside the system. Double-click on the NCD blocks to open the restriction NCD images and to display all the other restrictions. It is noted that restrictions for the first result define the limits step response as shown above. Meanwhile, the output restrictions limits keep the signal near the ± 0.05 value.

Before starting the optimization, the Optimization Parameters dialog box opens by selecting Parameters from the Optimization menu and it should be noticed that 'how' is defined by

the Parameters Optimization. Also opened is the Uncertain Variables and one observes 'how' is defined by the A matrix installation and that optimization restrictions are only nominally applied on installation.

One can press Start or one can hold down the acceleration key and start the optimizing process. Then, follows the time response evolution, during the optimizing process. Optimization time, cost function evolution and final values for tuneable parameters vary depending on the computer. Optimization should produce a regulator that satisfies all restrictions (Franklin, G., et al., 1987; Potvin, A.F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

Now, one returns to the Uncertainty Variables dialog box and one restricts the lower and upper limits. Press Start to begin optimizing with uncertainty. We can consider that these restrictions cannot be satisfied, since the result appears to show a maximum restraint violation of less than 0.01.

We can experiment by moving the restriction limits to achieve a higher system performance. For example, one reduces the propagation time or lessens the oscillation restrictions (Figure 10).

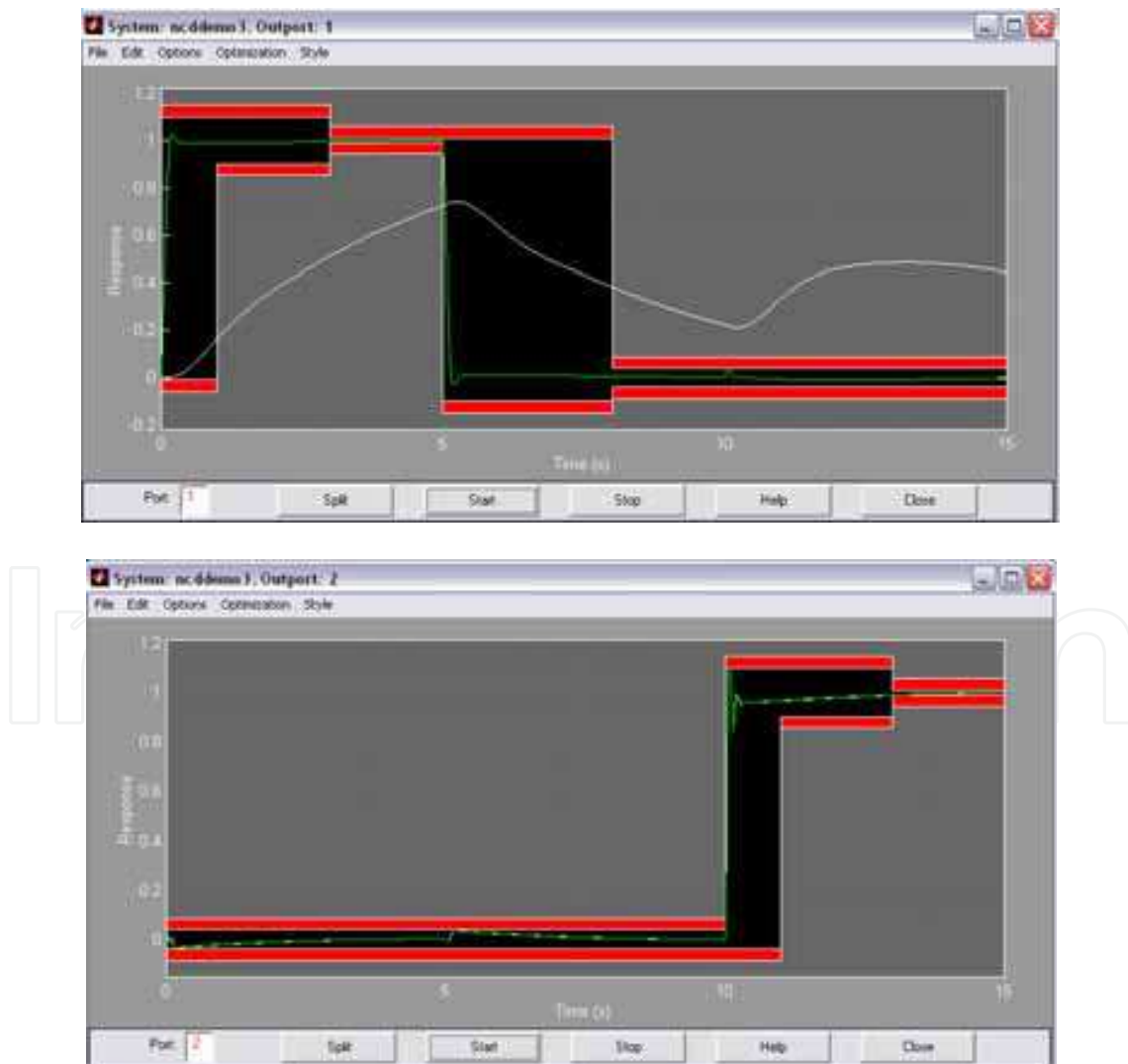


Fig. 10. Moving the limits of restriction to achieve better performance of the system

5. Case studies

We have chosen to present two case studies. The first is the inverted pendulum and the second is the modelling of a robotic arm.

5.1 Inverted pendulum

The inverted pendulum can be described as a cylindrical metal rod attached to a device controlled by an engine powered to revolve only around an axis. The device follows a linear track to create a stabilized problem (Figure 11).

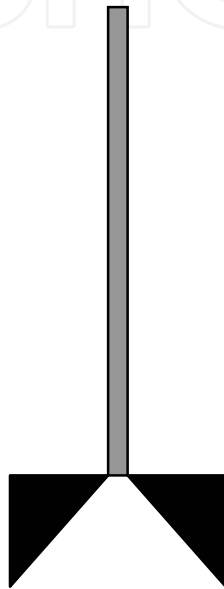


Fig. 11. Inverted pendulum

The inverted pendulum is a classic problem in dynamics and control theory, and is widely used as a benchmark for testing control algorithms (PID controllers, neural networks, fuzzy control, genetic algorithms, etc). Variations on this problem include multiple links, allowing the motion of the cart to be commanded while maintaining the pendulum and balancing the cart-pendulum system on a see-saw. The inverted pendulum is related to rocket or missile guidance, where thrust is actuated at the bottom of a tall vehicle. The understanding of a similar problem is built in the technology of Segway, a self-balancing transportation device. The largest implemented use of this is on huge lifting cranes in shipyards. When moving the shipping containers back and forth, the cranes move the box accordingly so that it never swings or sways. It always stays perfectly positioned under the operator, even when moving or stopping quickly.

Another way that an inverted pendulum may be stabilized, without any feedback or control mechanism, is by oscillating the support rapidly up and down. If the oscillation is sufficiently strong (in terms of its acceleration and amplitude) then the inverted pendulum can recover from perturbations in a strikingly counterintuitive manner. If the driving point moves in simple harmonic motion, the pendulum's motion is described by the Mathieu equation.

In practice, the inverted pendulum is frequently made of an aluminium strip, mounted on a ball-bearing pivot; the oscillatory force is conveniently applied with a jigsaw.

The equation of motion is similar to that for an un-inverted pendulum, except that the sign of the angular position is measured from the vertical unstable equilibrium position:

$$\ddot{\theta} - \frac{g}{l} \sin \theta = 0 \quad (4)$$

When added to both sides, it will have the same sign as the angular acceleration term:

$$\ddot{\theta} = \frac{g}{l} \sin \theta \quad (5)$$

Thus, the inverted pendulum will accelerate away from the vertical unstable equilibrium in the direction initially displaced and the acceleration is inversely proportional to the length. Tall pendulums fall more slowly than short ones.

The inverted pendulum with small parametric forcing is considered as an example of a wider class of parametrically forced Hamiltonian systems. The qualitative dynamics of the Poincare map corresponding to the central periodic solution is studied via an approximating integrable normal form. At bifurcation points we construct local universal models in the appropriate symmetry context, using the equi-variant singularity theory. In this context, structural stability can be proved under generic conditions (Van Noort, M., 2001).

The upper equilibrium of a pendulum can be stabilized by a vertical oscillation of the suspension point within a specific range of excitation frequencies and amplitudes. This follows from classical perturbation theory applied to the linearized equation of motion, e.g., according to van der Pol and Strutt, Stoker and Hale.

The corresponding bifurcation is determined by the non-linear dynamics. Our aim is to understand this dynamics in a qualitative way, with special interest in persistence.

Here, the symmetries of the system are first maintained. However, we consider a system that is slightly more general, but still in the 1 and ½ degree-of-freedom Hamiltonian setting (Van Noort, M., 2001).

We study the corresponding Poincare map, following the approach of Broer and Vegter.

Normal form theory yields a planar Hamiltonian vector field which gives an integrable approximation of this map, valid for every angular displacement and small velocity of the pendulum. The relation between the Poincare map and its approximation is briefly discussed in terms of perturbation theory (Sultan, K., 2003, 2007; Van Noort, M., 2001).

At each bifurcation point of the approximating vector field a model is constructed that is locally equivalent to this approximation, by performing small perturbations that respect the symmetries and conjugating these perturbations to the model by symmetry-preserving local morphisms (Van Noort, M., 2001).

The study of the inverted pendulum could be a collection of MATLAB functions and scripts, and SIMULINK models, useful for analyzing the inverted pendulum system and designing a control system for it.

The inverted pendulum is one of the most important classical problems of control engineering. 'Broom balancing' is a well known example of a non-linear, unstable control problem. This problem becomes further complicated when a flexible broom, in place of a rigid broom, is employed. The degree of complexity and difficulty in its control increases with its flexibility. This problem has been a research interest of control engineers. In this paper, however, we have analyzed the inverted pendulum only with a rigid broom (Sultan, K., 2003, 2007).

The aim of the study is to stabilize the inverted pendulum so that the position of the carriage on the track is controlled quickly and accurately so that the pendulum is always erected in its inverted position during such movements.

The inverted pendulum (IP) is among the most difficult systems to control in the field of control engineering. Due to its importance in the field of control engineering, it has been a task of choice to be assigned to control engineering students to analyze its model and propose a linear compensator according to the PID control law (Sultan, K., 2003, 2007).

For device, a stabilized runway requires the existence of the initial LQR stabilizer. The generation of this regulator rules is realized starting with writing the non-linear equations which define the inverted pendulum. Ignoring the dynamics of the engine, the non-linear equations of motion, for the inverted pendulum system, are (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003):

$$\ddot{y} = \frac{\frac{f}{m} + l\theta \sin \theta - g \sin \theta \cos \theta}{\frac{M}{m} + \sin^2 \theta} \quad (6)$$

$$\ddot{\theta} = \frac{-\frac{f}{m} \cos \theta + \frac{M+m}{m} g \sin \theta - l\theta^2 \sin \theta \cos \theta}{l \left(\frac{M}{m} + \sin^2 \theta \right)} \quad (7)$$

Where:

- f is the force applied to the cart by the engine in Newton (N);
- m is the position of the cart in meters;
- y is the vertical angle of the pendulum in radians;
- θ is the mass of the cart (0.45 kg);
- M is the pendulum mass (0.21 kg);
- l is the distance from the mass centre of the pendulum (half of its length of 0.61 m);
- g is gravitational acceleration (m/s).

It is necessary that those equations to be linear in the operating point $y=0$ and $\theta=0$ to obtain the linear system:

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{gm}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(M+m)}{lM} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{11}{M} \\ 0 \\ -\frac{1}{lM} \end{bmatrix} f = Ax + bx \quad (8)$$

Using MATLAB command:

$$K1qr = lqr(A,b,diag([0.25 \ 0 \ 4 \ 0],0.003)$$

one obtains the gained stability:

$$K1qr = (-28.86 \ -28.56 \ -145.00 \ 14 \ -14.86)$$

Besides the obvious non-linearity of the system's equations, the voltage limit applied to the engine gives a restriction of the action saturation of 1 N (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

The sensors measure the position and the angle of the pendulum device. In addition, according with the pendulum stability, one obtains a commanded reference signal for the presenting device. Specifically, one has to design a controller for the system to meet the following specifications of closed loop:

- up to 4 sec. for propagation time;
- up to 6 sec. for the response time;
- zero oscillation;
- less than 0.2 radians deviation from vertical.

It should be noted that the saturation non-linearity included in the process model and hidden pendulum block contain the motion system with non-linear equations.

One commands the input device position with a signal-type unit. NCD blocks are attached to the pendulum angle and device position signal (Figure 12). It is noted that each simulation lasts 15 sec.

The control structure contents have finite estimated status differences for the speed of the device and angular velocity of the pendulum.

As part of an internal control cycle, the estimations for the speed and angular velocity are multiplied by the amplification, collected and then introduced into the engine (Franklin, G., et al., 1987; Potvin, A. F., 1991; National Instruments; MathWorks; MathWorks User's Guide, 2003).

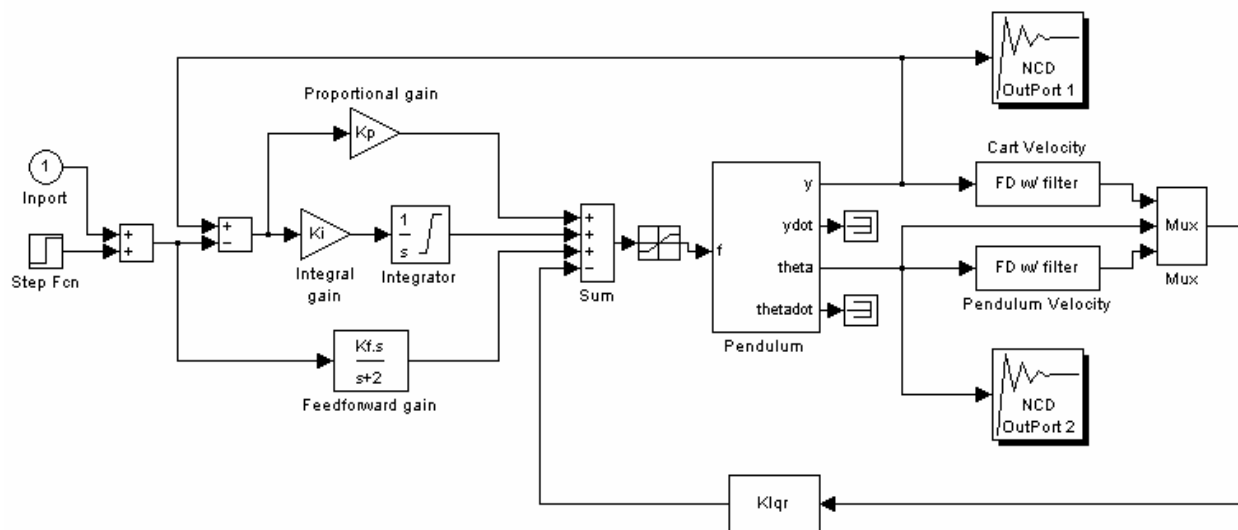


Fig. 12. NCD block attached to the result of the inverted pendulum

The gain is initialized from 1 at 3 with: $Klqr = Clqr(2:4)$, where $Clqr$ is the solution from 1 to 4 at LQR. In a series of external controls (used to allow the device to follow a commanded signal) a feed-forward amplifier Kf is initialized by $Kf=C1qr(1)$ and an integral amplifier, Ki is initialized as zero. It is noted that, in the absence of a commanded signal, these baseline controllers reduce the control structure at LQR amplification described in the previous section (Andrei, H., et al., 1999; Călin, S. et al., 1979; Franklin, G., et al., 1987; Potvin, A. F., 1991).

Tunable variables are initialized and, in accordance, restrictions of time domain response are defined. The configuration of these restrictions should be known like a step response. Pendulum angle channel contains restrictions that define a disturbance rejection problem of

perturbations. In other words, while the device moves to the controlled position, the pendulum must remain in balance.

The optimization begins. The optimization time, the cost function evolution and final values for tunable parameters vary depending on computer. Optimization should produce a controller to meet any restrictions.

It is noted that this optimization runs slower than others. This is, due to the estimating finite-state involved, frequently updated during the simulation (Figure 13).

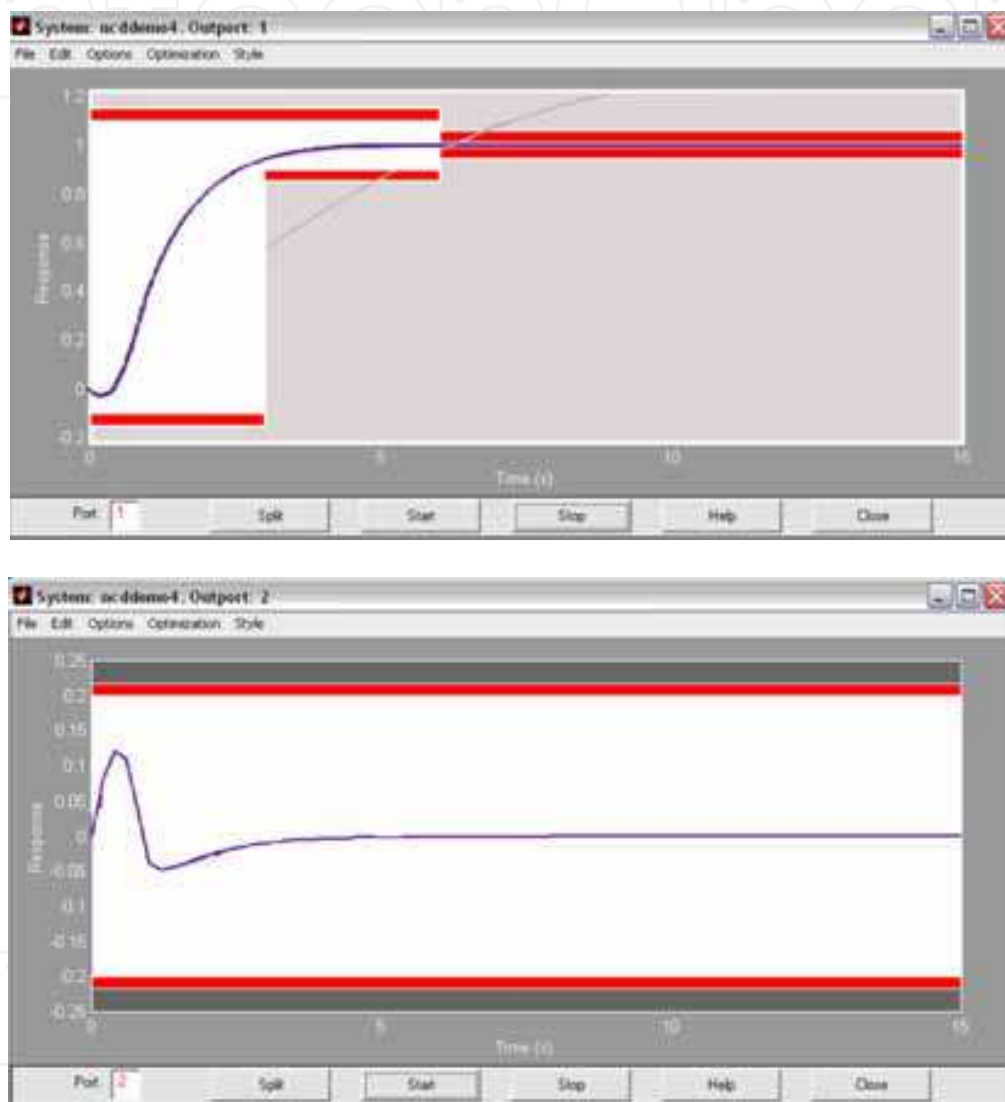


Fig. 13. Moving the limits of restriction to achieve better performance of the inverted pendulum

5.2 Modelling of a robotic arm

Modelling the manipulator's arm in a MATLAB-Simulink environment will be achieved by integrating the equations of state, twice. Note that the expressions depend, beside the mechanical parameters of the system, on the state variable and its first order derivative, which means considering them as integration reaction in the Simulink scheme of the two variables (Sciavicco, L., Siciliano, B., 2000; Stanciulescu F., 2003).

To obtain a general model that can be used regardless of the values of mechanical parameters of the system, the Simulink model will be developed with formal parameters. But, before starting the simulation, the numerical values of mechanical parameters (m_1 , m_2 , I_2 , b , g_0) will be initialized in the MATLAB environment. The following scheme is done in Simulink (Patic, P. C., Gorghiu, G., 2009; Stanciulescu, F., 2003):

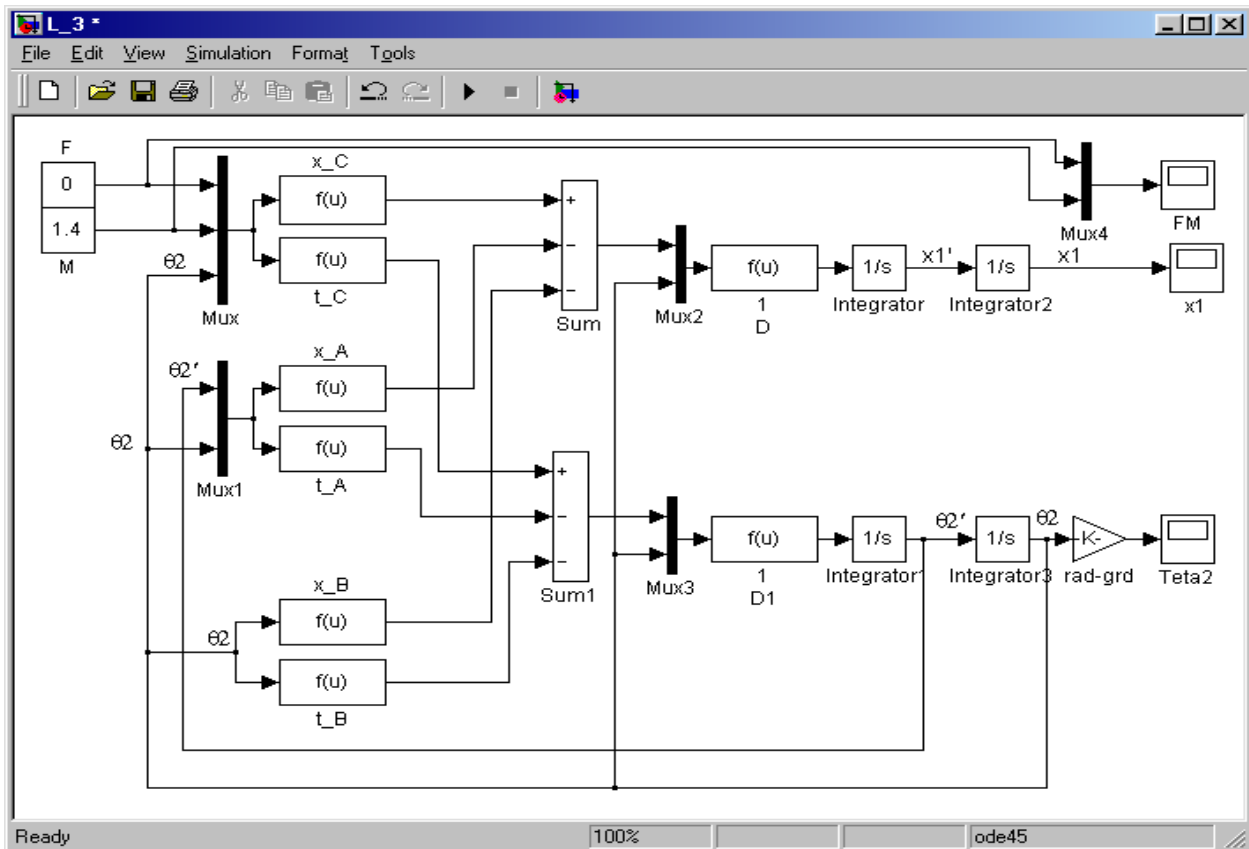


Fig. 14. Simulink scheme regarding the arm modelling

The localization of the blocks in Simulink sub-libraries is (Patic, P. C., Gorghiu, G., 2009; Stanciulescu, F., 2003):

x_A , x_B , x_C , $1/D$, t_A , t_B , t_C , $1/D1$ - blocks Fcn in Functions & Tables;

F, M - blocks Constant in Sources;

Sum, Sum1 - blocks Sum in Math;

Mux, Mux1 - blocks Mux in Signals&Systems;

Integrator, Integrator1, - blocks Integrator in Continuous;

Rad-grd - block Gain in Math;

FM, x1, Teta2 - blocks Scope in Sinks

The functions performed by each block are:

x_A : $-(I_2+m_2*b^2*(\sin(u[2]))^2)*m_2*b*\sin(u[2])*(u[1]^2)$;

x_B : $-(m_2*b)^2*\sin(u[1])*cos(u[1])*g_0$;

x_C : $(I_2+m_2*b^2*(\sin(u[3]))^2)*u[1]-(m_2*b*cos(u[3]))*u[2]$;

t_A : $(m_2*b)^2*\sin(u[2])*cos(u[2])*(u[1]^2)$;

t_B : $(m_1+m_2)*m_2*b*\sin(u[1])*g_0$;

t_C : $-(m_2*b*cos(u[3]))*u[1]+(m_1+m_2)*u[2]$;

$1/D$, $1/D1$: $u[1]/((m_1+m_2)*(I_2+m_2*b^2*(\sin(u[2]))^2)-(m_2*b*cos(u[2]))^2)$.

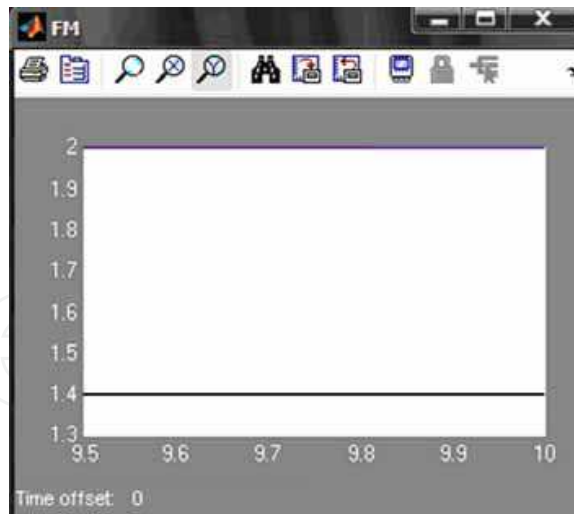


Fig. 15. The evolution of the system - changing external stimuli

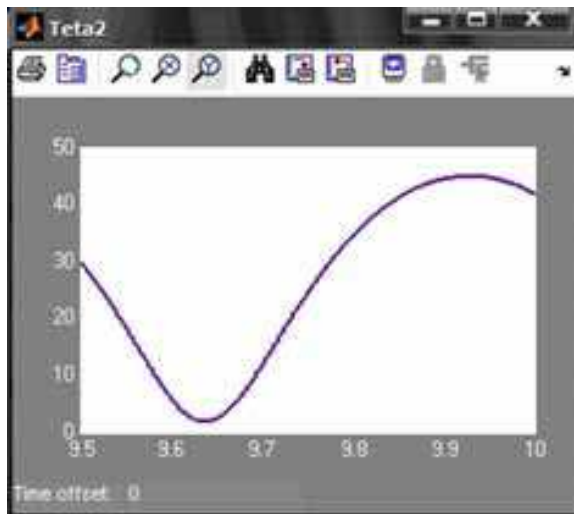


Fig. 16. The evolution of the system - changing external stimuli

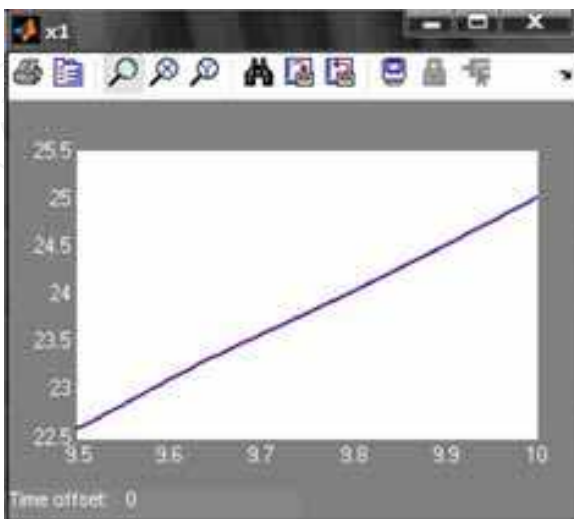


Fig. 17. The evolution of the system - changing external stimuli

The block rad-grd being Gain type, with value $180/\pi$, transform the position θ_2 from radians to degrees, just for viewing. As a method of integration, we will choose the method with variable step ode45, maximum integration step is imposed being 0.0001 [s], as well as the final time (stop time), which is 10 [s]. The maximum step 0.0001 was chosen to track the influence of changing values in the simulation stimuli on the evolution of the system (Stanciulescu, F., 2003; Starețu, I., Ionescu, M., 2005).

After the model realization, a MATLAB file by "m" type is created in which are initialized the values of the mechanical parameters (Patic, P.C., Gorghiu, G., 2009):

```
m1=1;
m2=1;
I2=0.01;
b=0.2;
g0=9.81;
brat;
```

When typing in MATLAB window, the name of this file will be loaded in MATLAB space with the mechanic's parameter values, the last line of the file causing the opening model (Stanciulescu, F., 2003).

The simulation will aim to highlight the evolution of the system to changes in external stimuli.

The results exemplified correspond to a force application $F = 2$ [Nm] at time of $t = 1$ sec.

5.1.1 Simulation of the robotic arm using the 3rd degree polynomial interpolation

Continuing our study, for the proper representation of variations of positions, velocities and accelerations for these study couplers robots, we used two methods, utilized very often, to solve these problems of movements of a robot. The first method used was the 3rd degree polynomial interpolation. The other method is the connection of linear functions in parables (Pozna, C., 2000).

One can say that both methods transform the task space in joint coordinates. Polynomials can be used to approximate more complicated curves. A relevant application is the evaluation of the natural logarithm and trigonometric functions. This results in significantly faster computations. Polynomial interpolation also forms the basis for algorithms in numerical quadrature and numerical ordinary differential equations.

Polynomial interpolation is also essential to perform sub-quadratic multiplication and squaring, where an interpolation through points on a polynomial which defines the product, yields the product itself (Scritube Industrial Robots).

The interpolation polynomial of three degree is like below:

$$P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad (9)$$

The statement that P interpolates the data points means that:

$$P(x_i) = y_i, \text{ where } i \in \{0, 1, 2, 3\} \quad (10)$$

The second method is the connecting of the linear functions in parables. This method is generally used when the robot trajectory passes through several points, but can be used, also, to move from point to point. To join two different points, rectilinear trajectories are used and to respect the conditions of speed and acceleration (by crossing points) straight paths are connected in parables.

The trajectory of motion of the robot is studied from two points of the workspace (Patic, P. C., Gorghiu, G., 2009).

Below we represent only the polynomial interpolation of degree 3 method, with a view for further research to developing the other method - connecting linear functions in parables. So one can observe, graphically, the methods explained, theoretically, above.

First of all, we choose to determine a simulation of a robot arm translation (T) which moves from P1 (0, 0, 0) to P2 (150, 0, 200) (the shape of the trajectory is not required). In fact, in the next case:

the translation couple make a displacement by 150 mm:

$$d_i=0, d_f=150 \text{ mm.}$$

the rotation couple make a rotation by 200 rad:

$$q_i=0, q_f=200 \text{ rad.}$$

So, one tries to move the robotic arm from point P1(0, 0, 0) to P2(150, 0, 200), using two couples with two different degrees of freedom - Translation and Rotation.

After the graphic representation one obtains that both actions, the accelerations and breakings, are different from one couple to another, because the lengths of paths, which are moving, are different. The movement is performed simultaneously on all three directions (Pozna, C., 2000).

In figure 18 one represents the displacement versus time. The displacement from couple 1 is smaller than those from couple 2 (Patic, P. C., et al., 2010).

In figure 19 one determines the velocities versus time. Like above, the speed for couple 1, is smaller than for couple 2.

The graphical differences are more evident than above, when one uses the displacements.

In figure 20 one has the accelerations versus time. The acceleration has a decreasing trend, being noted that in five seconds the accelerations for those two couples is zero (Patic, P. C., et al., 2010).

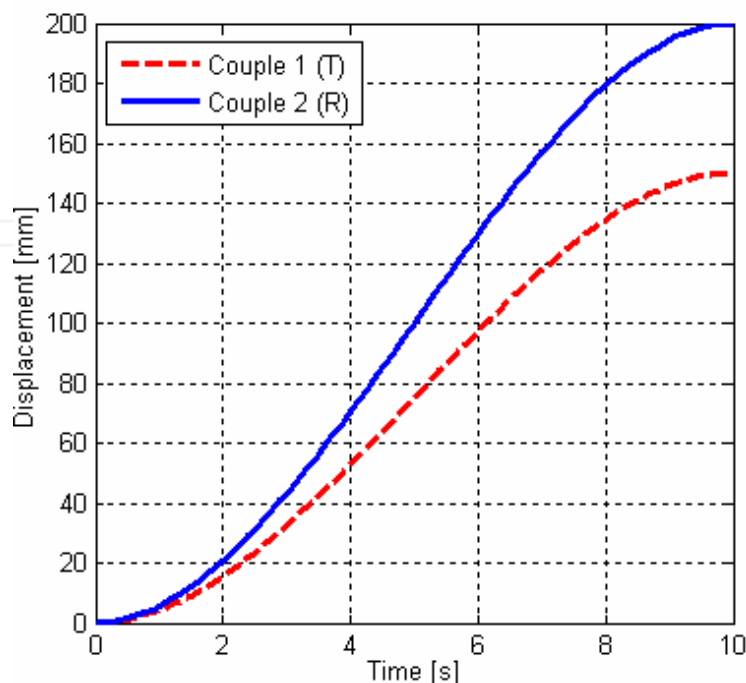


Fig. 18. The displacements of the robotic arm

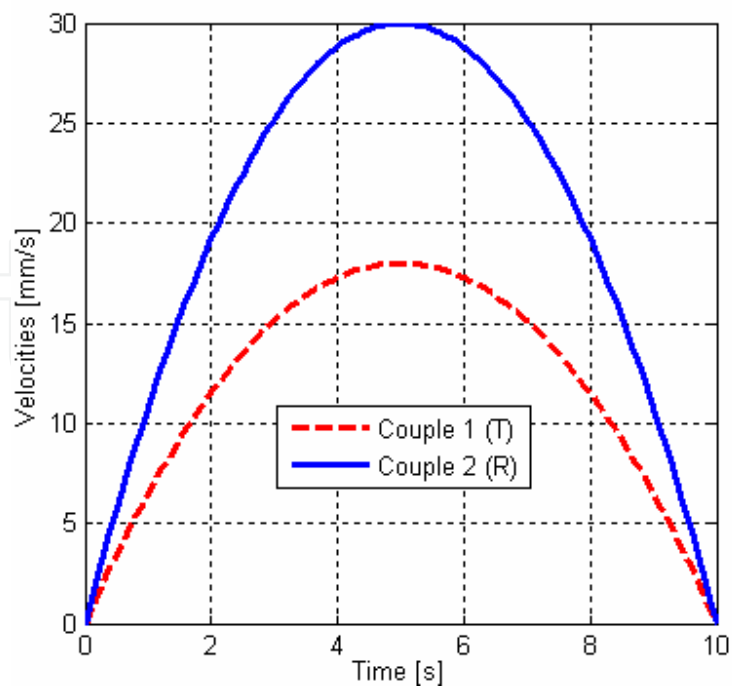


Fig. 19. The velocities of the robotic arm

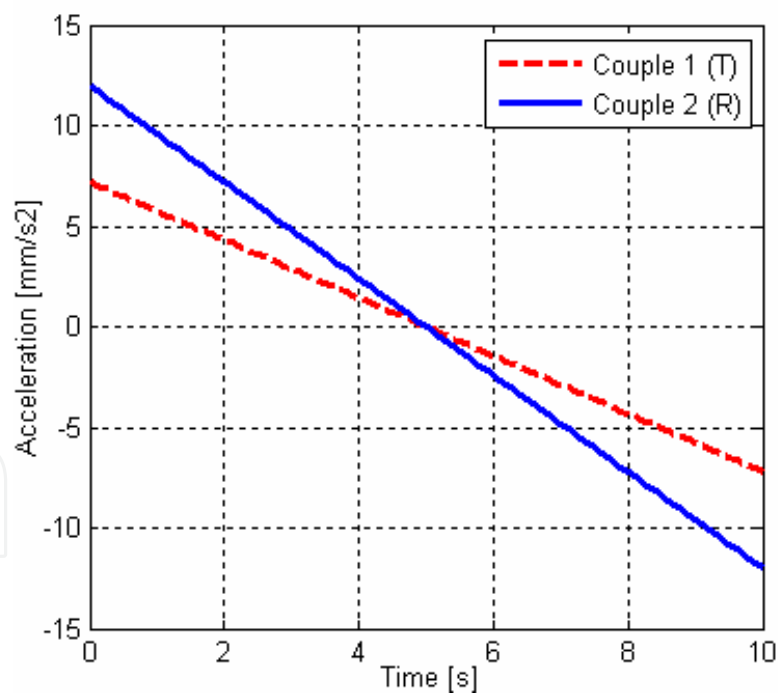


Fig. 20. The accelerations of the robotic arm

6. Conclusion

The work from this paper intended to determine the parameters of a PID controller using two software packages, namely MATLAB-Simulink and LabVIEW. As a mathematical model, a DC machine was used.

Automatic modelling systems using MATLAB-Simulink software packages applied the Cohen-Coon method for determining the closed loop PID parameters and plotted the system response curve. The limit of the over adjustment was found to be of 20%, the timing of the transitional regime - 3 seconds, and maximum response time was 10 seconds. Also, one represented the chart of the evolution over time, in a closed loop system, to unit step input. The use of these software packages offers an easy way of working, which means that it is convenient to utilize the programme functions.

Implementing the same system in a closed loop PID adjustment using the LabVIEW software package, one requires, at first, an additional procedure, namely, the use of the transfer functions in Z domain. For the displayed example the sampling period is 60 μ s and so can be determined the transfer function graph, sampled to the entry with a unitary step.

By the use of the graphical features of the LabVIEW software package for the adjustment of the PID controller, one recorded the best performance, since they include zero stationary error due to the integrative component for minimum stabilization time, due to the anticipatory component derivative etc.

Using the LabVIEW package to study the reaction systems involved additional procedures.

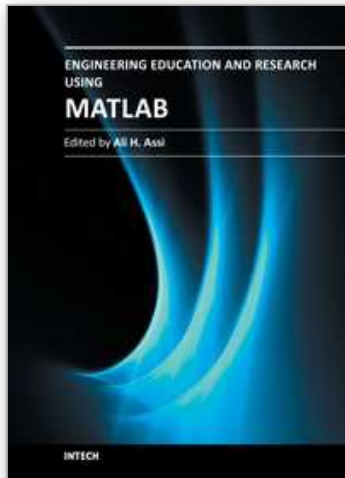
A comparative analysis of the two programmes using MATLAB-Simulink and LabVIEW to determine the same automatic performances of the system indicates that MATLAB-Simulink software is more efficient, having different features of its libraries already implemented, with coverage areas of automation and systems theory. LabVIEW is a software package dedicated to particular virtual instrumentation and graphical applications in time.

For the robotic simulation one can say that a functional simulation, using a 3D model is very important for obtaining an optimum version of a robotic arm and for a robot. After the graphic representation, one obtains that both actions, the accelerations and breakings, are different from one couple to another, because the lengths of paths, which are moving, are different. The movement is performed simultaneously on all three directions. In conclusion, one recorded the maximum values of the couples that make big displacements, so the engines that lead such couples are very loaded.

7. References

- Andrei, H., Brezeanu, I., Enciu, G., Enciu, M., *Elements of Adjustment Theory*, Ed. Macarie, Târgoviște, 1999
- Călin, S., Tertişco, M., Dumitrache, I., Popeea, C., Popescu, D., *Optimisation in Industrial Automation*, Ed. Tehnică, Bucharest, 1979
- Călin, S., Brezeanu, I., Popescu, M., Ispas, G., *Automatic Tuning System Theory. MATLAB Application*, Ed. Biblioteca, Târgoviște, 2002
- Franklin, G. F., David Powell, J. and Emami-Naeini, A., *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, 1987
- Dumitrache, I., *State-space Method for Linear Systems Automatic Analysis, Continuous and Discrete*, U.P.B. Litography, Bucharest, 1971
- Gibson, J., *Linear Automatic System*, English translation, Ed. Tehnică, Bucharest, 1967
- Ibrahim, H.E.A., *LQR Optimal Control for Micro-machined Tunnelling Accelerometer*, Proceedings of the 9th WSEAS International Conference on Signal Processing, Robotics and Automation, ISPR '10, February 20-22, Cambridge, U.K., p.40-45, 2010.
- Ionescu, V., *Introduction in Structural Theory of a Linear Systems*, Academy Edition., Bucharest, 1975

- Jalili-Kharaajoo, M., Dehestani, A., Active Queue Management: Comparison of Sliding Mode Controller and Linear Quadratic Regulator, *WSEAS Transactions on Circuits and Systems*, Issue 3, Volume 2, July 2003, ISSN 1109-2734, p. 594-599
- Sultan, K., Inverted Pendulum, MATLAB Central, 2003 and 2007, Available from <http://www.mathworks.com/MATLABcentral/fileexchange/3790-inverted-pendulum>
- Chugani, M. L., Samant, A. R., Cerna, M., *LabVIEW Signal Processing*, Pearson, 1998
- MathWorks, Using MATLAB-Simulink, n.d., Available from http://www.mathworks.com/access/helpdesk_r13/help/toolbox/ncd/case.html
- Mejhed, N. H., Hmamed, A., Benzaouia, A., Continuous Time Regulator for Linear Systems with Constrained Control, *WSEAS Transactions on Systems*, Issue 5, Volume 4, May 2005, ISSN 1109-2777, p. 537-545.
- National Instrument, NI Developer Zone, n.d., Available from <http://zone.ni.com/devzone/conceptd.nsf/webmain/2F266590B802FEFD8625686500609588/AN039.pdf>
- Nițu, C., Jora, B., Popeea, C., Georgescu, E., *Automatic Regulators. Laboratory Guide*, UPB, Bucharest, 1974
- Non-linear Control Design Blockset, for use with Simulink - User's Guide - The MathWorks, 2003
- Papadache, I., *Choosing and According of the Regulators*, Ed. Tehnică, Bucharest, 1975
- Patic, P. C., Pascale, L., Popa, F., Ardeleanu, M., *Modelling and Simulation of a Column Industrial Robot Type Used in Mounting Processes*, 10th Wseas International Conference on Signal Processing, Computational Geometry and Artificial Vision (ISCGAV '10), Taipei, Taiwan, August 20-22, 2010, ISBN 978-960-474-217-2, ISSN 1792-4618, pp. 122-127
- Patic, P. C., Gorghiu G., *The Modelling and Simulation of a Robotic Arm*, *The Scientific Bulletin of Electrical Engineering Faculty*, ISSN 1843-6188, 2009, No. 2 (11), pp. 85 - 90
- Potvin, A. F., *A Unified Solution to Constrained Configuration Control Law Design*, Master's Thesis, MIT EECS Dept., 1991
- Pozna, C., *Command and Control of the Industrial Robots*, CIT Publishing House, Brasov 2000
- Bishop, R., *LabVIEW 2009 Student Edition*, Prentice Hall, 2010
- Roshan N., *Application of Linear Quadratic Regulator (LQR) in Displacement Control of an Active Mass Damper*, 4th WSEAS International Conference on Systems Theory And Scientific Computation (ISTASC'04), December, 17-19 2004, Tenerife, Spain.
- Savant, C. J., *Automatic System Calculus*, Ed. Tehnică, Bucharest, 1967
- Sciavicco, L., Siciliano, B., *Modelling and control of robot manipulators*, Springer, 2000
- Scritube Industrial Robots, n.d., Available from <http://www.scritube.com/tehnica-mecanica/Roboti-Industriali-Introducere45272.php>
- Călin, S., *Automatic Regulators*, Ed. Didactică și Pedagogică, Bucharest, 1970
- Sprânceană, N., Dobrescu, R., Borangiu Th., *Discrete Automation in Industry*, Ed. Tehnică, Bucharest, 1978
- Stanculescu, F., *The Complex System Modelling*, Ed. Tehnică, 2003
- Starețu, I., Ionescu, M., Kinematics and Functional Simulation of a Robotic Arm from a Pyrotechnic Robot, 2005, Available from http://www.imsar.ro/SISOM_Papers_2005/1_R.pdf
- Van Noort, M., Dissertaties - Rijksuniversiteit Groningen, 2001, Available from <http://dissertations.ub.rug.nl/FILES/faculties/science/2001/m.van.noort/c2.pdf>



Engineering Education and Research Using MATLAB

Edited by Dr. Ali Assi

ISBN 978-953-307-656-0

Hard cover, 480 pages

Publisher InTech

Published online 10, October, 2011

Published in print edition October, 2011

MATLAB is a software package used primarily in the field of engineering for signal processing, numerical data analysis, modeling, programming, simulation, and computer graphic visualization. In the last few years, it has become widely accepted as an efficient tool, and, therefore, its use has significantly increased in scientific communities and academic institutions. This book consists of 20 chapters presenting research works using MATLAB tools. Chapters include techniques for programming and developing Graphical User Interfaces (GUIs), dynamic systems, electric machines, signal and image processing, power electronics, mixed signal circuits, genetic programming, digital watermarking, control systems, time-series regression modeling, and artificial neural networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Patic Paul Ciprian, Duta Luminita and Pascale Lucia (2011). Control Optimization Using MATLAB, Engineering Education and Research Using MATLAB, Dr. Ali Assi (Ed.), ISBN: 978-953-307-656-0, InTech, Available from: <http://www.intechopen.com/books/engineering-education-and-research-using-matlab/control-optimization-using-matlab>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen